

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 58-053010

(43)Date of publication of application : 29.03.1983

---

(51)Int.Cl. G11B 5/09  
G06F 13/04

---

(21)Application number : 56-149523 (71)Applicant : HITACHI LTD

(22)Date of filing : 24.09.1981 (72)Inventor : INOUE KATSU  
TSUYUKI HISAMASA

---

## (54) SERIAL RECORDING DATA FILE CONSTITUTING METHOD

### (57)Abstract:

PURPOSE: To reduce the running distance of a recording medium by recording an index representing the content of each record in a data file that is a header in the final record of the data file.

CONSTITUTION: The final records of files 20W90 are referred to as index records 293989 and 99 where a value B of data records recorded with the newest data is recorded. Further a value A representing a file which is in use at present for the 1st record 11 only is recorded in an index 10 in the same way as conventional methods. The value B is written once when the data file is completed without being recorded on every occasion.

⑨ 日本国特許庁 (JP)

⑩ 特許出願公開

⑫ 公開特許公報 (A)

昭58—53010

⑪ Int. Cl.<sup>3</sup>

G 11 B 5/09

G 06 F 13/04

識別記号

1 0 1

庁内整理番号

7629—5D

7361—5B

⑬ 公開 昭和58年(1983)3月29日

発明の数 1

審査請求 未請求

(全 3 頁)

⑭ シリアル記録データファイル構成法

⑯ 発明者 露木寿正

勝田市市毛882番地株式会社日立製作所那珂工場内

⑰ 特 願 昭56—149523

⑱ 出 願 昭56(1981)9月24日

⑲ 出 願 人 株式会社日立製作所

⑳ 発 明 者 井上克

勝田市市毛882番地株式会社日立製作所那珂工場内

東京都千代田区丸の内1丁目5番1号

㉑ 代 理 人 弁理士 高橋明夫

明 細 書

発明の名称 シリアル記録データファイル構成法

特許請求の範囲

1. シリアル記録方式のデータファイルにおいて、複数のデータファイルのファイル名、データ記録に使用しているレコード数などを総括した1個のインデックスファイルの他に、各データファイルの各レコードの内容を総括する第2のインデックスを、該データファイルの最終レコードに持たせたことを特徴とするシリアル記録データファイル構成法。

発明の詳細な説明

本発明の対象は、小形卓上電子計算機などで、毎日の作業記録を磁気テープなどに記録する場合のデータファイルの構成法である。

従来のシリアル記録データでは各ファイルの先頭にヘッダを設けて、該1個のファイルの内容を知ることができるか、ファイルの先頭に設けられていた。一方、簡単なファイル構成では、各フ

イルにはヘッダを設けず1個のインデックスにこれらの情報を総括する場合があるが、この方式でヘッダを追加することは困難であつた。それは既に無ヘッダで先頭から、データが記録されているのである。しかるに、1個のインデックスのみの簡単な構成では、テープ記録が進行し、インデックス、データ両ファイル間のテープ上の間隔が大きくなると、1レコード分記録毎のテープを走行距離が飛躍的に増加し、テープの寿命を短くすると云う欠点があつた。

本発明の目的は、かかる短寿命と考えられる簡易構成法の記録データを、データ変換することなく、そのままヘッダ付きのシリアル記録ファイル構成とするに、若干のプログラム変更と、データファイル構成の変更により達成せんとするものである。

そのため、本発明においては、一般的にはデータが記録されていない、データファイル中の最終レコードに、そのデータファイル中の各レコードの内容を示す第2のインデックス即ちヘッダを記

録し、各ファイルの参照、書き込み、読出しには他ファイルであるインデックスを読取ることなく、当該ファイルのみの読出しで可能とするものである。

第1図は磁気テープ100上にデータファイル20~90と、インデックスファイル10を構成した従来例で、今データファイル80の第nレコード83に、最新データが記録されているとする。インデックスファイル10の第1レコード11はファイル80を値Aが記録され、Aレコード18には第nレコードを示す値Bが記録されており、常に最新データの所在即ちデータファイル80の第nレコードが示されている。

一方第2図は本発明の構成で、ファイル20~90の最終レコード(第1図の25, 35, 85, 95)をインデックスレコード29, 39, 89, 99としたもので、そこには最新データが記録されているデータレコードの値Bが記録されている。一方インデックス10には、第1レコード11のみに現在使用中のファイルを示す値Aが従来と同

本発明を実現するプログラム例として、第3図の流れ図を示す。実行ステップ1, 2が従来のステップ3, 4に置き換えられたわけである。今、第2図に於て、データファイル80のレコード83に最終データが記録され、次のレコード84に新しいデータを記録せんとする場合を考える。最終レコード89から最終データが記録されているレコード83を示す値11Bを読み取り、Bを1増加させて、最終レコードを書き直し新データを記録する。

変形例として、レコード83を示す値は常時計算機の内部メモリに記憶しておき、記憶が失われた時のみ最終レコードから読取る如くし、先ずデータレコード84を記録してから、最終レコード89の内容即ちB+1に記録し直せば、データレコード記録中の事故に対しても、記録中のデータは失われるが、インデックス内容と不一致は生じない。

本発明のファイル構成法によれば、磁気テープなどシリアル記録媒体の走行距離を大幅に減らし

じく記録されている。又、Aレコード18には値Bはその都度記録されることなく、データファイルが完結した時点で一度だけ書込まれる。

この為、データの新レコードへの記録毎に、他ファイルであるインデックスに、値Bを記録する必要が無い為、インデックス、データファイル間の距離が大きくなる、データファイルの増えた際の、テープ走行距離を大幅に減すことができる。

今1データファイルのレコードサイズをm、データファイルの総数をnとすると、従来のインデックス方式では総計 $mn\{(n+1)+(m-1)^2\}$ の距離を、本案では $mn(n+1)$ の距離を走行するから、その走行距離は $1+(m-1)^2/(n+1)$ 分の1、即ち約 $n/m^2$ に減少する。n, mが100程度の場合には約 $1/100$ となりその効果は大きい。

勿論これ等の方法は、何時計算機の内部メモリの記録が失われても良い様に、常に外部メモリに記録することを考えている。従つて蒸発しない大量のバッファメモリが安価に入手できる場合は、かかるテープ走行は不要である。

得て且つ、従来のヘッダ無しファイル構成のデータを容易に矛盾なく変更できる。1データファイルがmレコードから成り、全部でnデータファイルを構成した場合そのテープ走行距離は $n/m^2$ に減少し、従つてアクセス時間も同様に短縮しテープ寿命も $m^2/n$ 倍に伸びると考えられる。

本発明のファイル構成法の第2の特徴は既に従来形式でできあがつた。データを、各データの最終データをヘッダに充てることにより、貴重な初期データを失うことなく、又はデータの記録の番地変換することなく、ヘッダ付きデータファイルとすることができる。各ファイルの最初の部分はデータの少いファイルであつても必ずデータが記録されているか、最終レコードは、レコードサイズが充分の大きさに選ばれるべきものである以上、データが記録されていること、即ちmレコードの全部にデータが詰っている確率は極めて低いと見て良い。

図面の簡単な説明

第1図は従来のシリアル記録ファイル構成図、

特開昭58-53010 (3)

第2図は本発明のシリアル記録ファイル構成法説明図、第3図は本発明を実現するプログラムの流れである。

10…インデックスファイル、11…第1レコード、20…データファイル1、21…第1レコード、80…データファイルn-1、85…最終レコード、29、39、…99…最終レコードを用いたヘッダ、100…シリアル記録媒体。

代理人 弁理士 高橋明夫

